EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	15	(delta near4 encoding near4 header)	US-PGPUB; USPAT; EPO; JPO	OR	ON	2006/09/18 12:58
L2	27	("6765925" "6901049" "6909715" "6788707" "6594280" "6181716" "5646617" "5737733" "6889385" "6882637" "6535925" "5530645" "6078955").pn.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/09/18 14:20
L5	10	"20010004768" "20050030944" "20020191691" "20030185245" "20020029206"	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/09/18 14:22
L6	430	DOCSIS near4 network	US-PGPUB; USPAT; EPO; JPO	OR	ON	2006/09/18 14:22
L7	3295	delta near3 encod$4	US-PGPUB; USPAT; EPO; JPO	OR	ON	2006/09/18 14:22
L8	6	L6 and L7	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/09/18 14:23

Delta encoding

From Wikipedia, the free encyclopedia
(Redirected from Delta compression)

Delta encoding is a way of storing or transmitting data in the form of differences between sequential data rather than complete files. Delta encoding is sometimes called **delta compression**, particularly where archival histories of changes are required (e.g., in software projects).

The differences are recorded in discrete files called "deltas" or "diffs". Because changes are often small (only 2% total size on average), delta encoding greatly reduces data redundancy. Collections of unique deltas are substantially more space-efficient than their non-encoded equivalents.

(Delta encoding should not be confused with Elias delta coding).

Perhaps the simplest example is storing values of bytes as differences (deltas) between sequential values, rather than the values themselves. So, instead of 2, 4, 6, 9, 7, we would store 2, 2, 2, 3, -2. This is not very useful when used alone, but it can help further compression of data in which sequential values occur often. IFF 8SVX sound format applies this encoding to raw sound data before applying compression to it. Unfortunately, not even all 8-bit sound samples compress better when delta encoded, and the usability of delta encoding is even smaller for 16-bit and better samples. Therefore, compression algorithms often choose to delta encode only when the compression is better than without. However, in video compression delta frames can considerably reduce frame size, and are used in virtually every video compression codec.

A delta can be defined in 2 ways, *symmetric delta* and *directed delta.* A *symmetric delta* can be expressed as $\Delta(v_1, v_2) = (v_1 \setminus v_2) \cup (v_2 \setminus v_1)$, where v_1 and v_2 represent two successive versions.

A *directed delta*, also called a change, is a sequence of (elementary) change operations which, when applied to one version v_1, yields another version v_2 (note the correspondence to transaction logs in databases).

A variation of delta encoding which encodes differences between the prefixes or suffixes of strings is called incremental encoding. It is particularly effective for sorted lists with small differences between strings, such as a list of words from a dictionary.

In delta encoded transmission over a network where only a single copy of the file is available at each end of the communication channel special error control codes are used to detect which parts of the file has changed since its previous version.

The nature of the data to be encoded influences the effectiveness of a particular compression algorithm. Delta encoding performs best when data has small or constant variation; for an unsorted data set, there may be little to no compression possible with this method.

The following C code performs a simple form of delta encoding and decoding:

```
void delta_encode(char *buffer, int length)
{
  char t = 0;
  char original;
  int i;
  for(i = 0; i < length; i++)
  {
    original = buffer[i];
    buffer[i] -= t;
    t = original;
  }
}

void delta_decode(char *buffer, int length)
{
  char t = 0;
  int i;
  for(i = 0; i < length; i++)
  {
    buffer[i] += t;
    t = buffer[i];
  }
}
```

Another instance of use of delta encoding is RFC 3229, "Delta encoding in HTTP," where The Internet Society proposes that HTTP servers should be able to send updated Web pages in the form of differences between versions (deltas), which should decrease Internet traffic, as most pages change slowly over time, rather than being completely rewritten repeatedly:

> This document describes how delta encoding can be supported as a compatible extension to HTTP/1.1.
>
> Many HTTP (Hypertext Transport Protocol) requests cause the retrieval of slightly modified instances of resources for which the client already has a cache entry. Research has shown that such modifying updates are frequent, and that the modifications are typically much smaller than the actual entity. In such cases, HTTP would make more efficient use of network bandwidth if it could transfer a minimal description of the changes, rather than the entire new instance of the resource.

See also

- Algorithm
- Data compression
- Data structure
- Delta modulation
- Encoding
- List of algorithms
- List of delta encoding software
- String-to-string correction problem

External links

- RFC 3229 - Delta Encoding in HTTP
- RFC 3284 - The VCDIFF Generic Differencing and Compression Data Format

Retrieved from "http://en.wikipedia.org/wiki/Delta_encoding"

Categories: Lossless compression algorithms | Data compression

- This page was last modified 03:03, 25 August 2006.